

United States Patent Application

for

DIGITAL IMAGE APPEARANCE ENHANCEMENT AND  
COMPRESSIBILITY IMPROVEMENT METHOD AND SYSTEM

Inventors:

Clayton Brian Atkins  
Jay Stephen Gondek

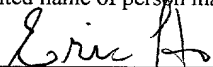
EXPRESS MAIL CERTIFICATE OF MAILING

"Express Mail" mailing label number EK910712915US

Date of Deposit March 7, 2001

I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Commissioner of Patents and Trademarks, Washington, D.C. 20231.

ERIC HO  
(Typed or printed name of person mailing paper or fee)

  
(Signature of person mailing paper or fee)

DIGITAL IMAGE APPEARANCE ENHANCEMENT AND COMPRESSIBILITY  
IMPROVEMENT METHOD AND SYSTEM

FIELD OF THE INVENTION

5           The present invention relates generally to image processing, and more particularly, to a digital image appearance enhancement and compressibility improvement method and system.

BACKGROUND OF THE INVENTION

10           Printers are well known peripherals that render text and images. With the advent and growth of the digital camera market, there is increased consumer demand for low-cost printers that render digital images well. Consequently, one design consideration for printer manufacturers is how to improve the perceived quality of the rendered images. Typically, printer manufacturers utilize software image processing programs to improve picture quality.

15           For example, image-sharpening algorithms are often used to improve the appearance of images. The intended effect of these image-sharpening algorithms is to increase pixel contrast on and around edges, which theoretically should lead to greater perception of details. Unfortunately, however, such image sharpening algorithms also enhance noise and compression artifacts, which can actually exacerbate image appearance. Furthermore, these  
20 image-sharpening algorithms reduce compressibility of the image, which is undesirable as described more fully hereinafter.

25           Other algorithms have been developed for improving the appearance of images by actively suppressing noise and artifacts. However, such algorithms have a smoothing effect on and around sharp edges, which leads to a softer appearance (i.e., a fuzzy image with less perception of details). Moreover, such algorithms can also be very computationally expensive, thereby requiring higher implementation costs and complexity, making these algorithms less attractive.

A second design consideration is how to decrease the time that a user has to wait from the time a print command is issued to the time the printing is completed. The latency (i.e., time that elapses between the time the print command is issued and the completed print job) is a function of the time that is needed to transfer the digital image from the PC to the printer and the time that is needed by the printer to actually render the image. The limiting factor is often the speed at which the image to be printed can be communicated from the computing appliance (e.g., personal computer (PC)) to the printer. For example, the speed of this communication is often limited by the bandwidth of the cable connecting the PC to the printer. In order to improve the speed of communication, one of many well-known compression algorithms are applied to the image so that the number of bits or data symbols that needs to be transferred between the PC and printer is decreased, thereby reducing the time needed to communicate information between the PC and the printer.

For example, one approach to improve compressibility is to apply a smoothing filter. The smoothing makes neighboring pixels more consistent with each other, which allows the image to be represented using fewer bytes in computer memory. Unfortunately, smoothing can degrade image appearance by reducing edge contrast. In other words, smoothing militates against the first design consideration of improving the image appearance.

Based on the foregoing, there remains a need for a method and system for improving the appearance of digital images while improving the compressibility of the image and that overcomes the disadvantages set forth previously.

### SUMMARY OF THE INVENTION

According to one embodiment of the present invention, a method and system for processing a digital image and improving the appearance of the image while enhancing the compressibility of the image are provided. The digital image has a plurality of input pixels.

In one embodiment, the image processing system has a filter selection mechanism for receiving a filter selection window corresponding to a current input pixel and responsive thereto for generating a filter identifier based on either one or more edge parameters computed based on the filter selection window or an activity metric computed based on the filter selection window. A filter application unit that is coupled to the filter selection mechanism for receiving the filter identifier and applying a filter identified by the filter identifier to an input pixel window to generate an output pixel is also provided. The filter selection window may be the same as the input pixel window.

According to another embodiment, the image processing method performs the following steps for each input pixel. First, an input pixel window that typically includes the current input pixel and pixels adjacent to the current input pixel is received. Second, a filter identifier is generated based on either an edge parameter computed based on the input pixel window or an activity metric computed based on the input pixel window. Third, a filter specified by the filter identifier is applied to the input pixel window to generate an output pixel corresponding to the current input pixel.

According to another embodiment, the image processing method performs the following steps for each input pixel. First, a level of activity is generated based on a first window of pixels with reference to the input pixel. Next, it is determined whether the level of variation is in a predetermined relationship with a predetermined level of variation. When the level of variation is in a predetermined relationship with a predetermined level of variation, the input pixel is replaced by a

blurred version of the input pixel. When the level of variation is not in a predetermined relationship with a predetermined level of variation, a measure of one or more edge parameters is generated based on a second window of pixels with reference to the input pixel. Then, an enhancement filter is selected based on the

5 measure of edge parameter and applied to the input pixel.

10004248-1

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings and in which like reference numerals refer to similar elements.

5        FIG. 1 is a block diagram illustrating an exemplary computer system in which the appearance enhancement and compressibility improvement mechanism of the present invention can be implemented.

10        FIG. 2 is a block diagram that illustrates in greater detail the appearance enhancement and compressibility improvement mechanism of FIG. 1 in accordance with one embodiment of the present invention.

15        FIG. 3 is a block diagram that illustrates an exemplary implementation of the appearance enhancement and compressibility improvement mechanism of FIG. 2.

20        FIG. 4 is a block diagram that illustrates in greater detail the edge dependent filter selection module of FIG. 3.

25        FIG. 5 is a flowchart illustrating the steps performed by the appearance enhancement and compressibility improvement mechanism in accordance with one embodiment of the present invention.

30        FIG. 6 is a flowchart illustrating the steps performed by the appearance enhancement and compressibility improvement mechanism in accordance with a preferred embodiment of the present invention.

35        FIG. 7 is a flowchart illustrating in greater detail the step of computing pixel components by utilizing a nonlinear filter of FIG. 6 in accordance with one embodiment of the present invention.

40        FIG. 8 is a flowchart illustrating the process by which a particular filter is selected in accordance with one embodiment of the present invention.

45        FIG. 9 illustrates a process by which edge angle coefficients are determined in accordance with one embodiment of the present invention.

FIG. 10 is a flowchart illustrating the process by which a class index is determined based on the edge angle coefficients in accordance with one embodiment of the present invention.

FIG. 11 illustrates a process by which curvature coefficients are determined in  
5 accordance with one embodiment of the present invention.

FIG. 10 is a flowchart illustrating the process by which a class index is determined based on the edge angle coefficients in accordance with one embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

A method and system for processing digital images and improving the appearance of the images while enhancing the compressibility of the images are described. In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, to one skilled in the art that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to avoid unnecessarily obscuring the present invention.

Computer System 100

FIG. 1 is a block diagram illustrating an exemplary computer system 100 in which the appearance enhancement and compressibility improvement mechanism 140 of the present invention can be implemented. The computer system 100 includes a personal computer (PC) 110 coupled to an office machine 120 via a cable 130. The PC 110 includes a processor 114 for executing software instructions and a driver program 118, which when executed by the processor 114, communicates and controls the office machine 120. For example, the driver program 118 can include software instructions for performing signal processing that is well known by those of ordinary skill in the art on digital information to be generated by the office machine 120.

The office machine 120 can be, but is not limited to, a printer, a copier, a scanner, an all-in-one office machine, and any device that renders images and/or text. For example, when the office machine 120 is a printer, the driver program 118 is a printer driver that has software instructions, which when executed by the processor 114, controls and manages the printing or rendering process for the printer.

The PC 110 can include the appearance enhancement and compressibility improvement mechanism (AECIM) 140 of the present invention. The AECIM 140 includes an appearance enhancement mechanism 150 for improving the appearance of the digital image and a compressibility enhancement mechanism 160 for improving the compressibility of the digital image. As noted previously, it is desirable to improve the compressibility of an



image in order to increase the speed at which a digital image can be communicated to the office machine 120.

The AECIM 140 includes an input for receiving an input digital image 144 and an output for generating a corresponding output digital image 148. As described in greater detail hereinafter with reference to FIG. 2, the AECIM 140 can receive an input pixel window, select a filter appropriate for a current pixel, and apply the selected filter to the input pixel window to generate an output pixel that replaces the current input pixel.

It is noted that the appearance enhancement and compressibility improvement mechanism 140 of the present invention can be implemented in software (e.g., in a driver), in firmware, in hardware, or in a combination thereof. For example, in one embodiment, the AECIM 140 can be implemented as software instructions as part of the driver program 118.

Moreover, the AECIM 140 of the present invention can reside in PC 110, office machine 120, or in a device that is disposed remote from the PC 100 and the office machine 120. For example, the AECIM 140 can be implemented as hardware circuitry that is disposed entirely in the PC 110 or entirely in the office machine 120. Alternatively, the AECIM 140 can be implemented as software that reside on a computer readable medium (e.g., a memory element, such as RAM or ROM, a computer disk, or compact disc) accessible by the PC 110 or the office machine 120. In another embodiment, the AECIM 140 can be in the form of software code that resides on a server that is part of a network (e.g., the Internet) to which PC 110 or office machine 120 is connected.

Alternatively in a distributed implementation, a first portion of the AECIM 140 of the present invention can reside in PC 110, a second portion of the AECIM 140 can reside in the office machine 120 and other portions of the AECIM 140 can be distributed in other devices, where the portions can be software, hardware, firmware or a combination thereof.

#### AECIM 140

FIG. 2 is a block diagram that illustrates in greater detail the appearance enhancement and compressibility improvement mechanism of FIG. 1 in accordance with one embodiment of the present invention. The AECIM 140 includes a filter selection mechanism 230 for

selecting an appropriate filter for use and a filter application mechanism 240 for applying the selected filter to the input pixel window 210 to generate an output pixel. The input pixel window is a plurality of pixels to which a filter is applied to generate a corresponding output pixel. It is noted that the input pixel window 210 typically includes the current input pixel.

- 5 However, the input pixel window 210 can include pixels about the current input pixel (e.g., a neighborhood of pixels adjacent to the current pixel) without including the current pixel.

The filter selection mechanism 230 selects a filter (e.g., a set of filter coefficients) from among a plurality of filters based on a filter selection window, which is this embodiment is the input pixel window 210. The filter selection window is a plurality of pixels that are used to select an appropriate filter. The filter selection mechanism 230 can select a filter appropriate for the current input pixel by using one or more factors.

For example, in one embodiment, the filter selection mechanism 230 employs an edge parameter evaluation unit 234 for computing an edge parameter corresponding to the input pixel window and utilizing the edge parameter to select an appropriate filter (e.g., a suitable set of filter coefficients for the input pixel window). Based on the input pixel window 210, the filter selection mechanism 230, for example, can select a blurring filter, a smoothing filter, a sharpening filter, or an enhancement filter based on one or more parameters computed from the input pixel window. As described in greater detail hereinafter, edge parameter can include any measurable unit that describes one or more traits or characteristics of an edge. The edge parameter can be, for example, an edge angle, edge sharpness, edge curvature, etc.

An edge is simply a contour to divide two regions. Edge angle is a characteristic of an edge that conveys the orientation of a contour between two regions. Edge sharpness is the width of a transition region between the two regions. For example, the edge sharpness can be the width of a transition region between two regions having different colors or intensity values. Edge sharpness can also be expressed as the rate change of the transition moving in a direction perpendicular to the edge angle. A sharper edge means that the transition region is narrow or less wide. A smooth edge means that the transition is wide or less narrow. Edge

curvature is a characteristic of an edge that conveys the rate change of sharpness as one moves toward the edge. For example, the edge curvature can convey whether the current side of the edge is light or dark.

In another embodiment, the filter selection mechanism 230 employs an activity metric evaluation unit 238 for computing a metric of activity in the input pixel window (e.g., the pixels in a neighborhood of the current pixel) and utilizing the activity metric to select an appropriate set of filter coefficients for the input pixel window. An example of an activity metric that can be utilized is a level of variation that is described in greater detail hereinafter with reference to FIG. 3.

It is noted that the filter selection mechanism 230 can select a filter (e.g., a set of filter coefficients) that are appropriate for the current input pixel by using an edge parameter, an activity metric, another measurable parameter, or a combination thereof.

It is further noted that the filter selection window employed to determine an activity metric can be different from or the same as the input pixel window 210. Similarly, the filter selection window employed to determine an edge parameter can be different from or the same as the input pixel window 210. Also, the filter selection window employed to determine an edge parameter can be different from or the same as the filter selection window employed to determine an activity metric.

It is to be understood that although a 5 x 5 square window of pixels is used in the example described herein, the input pixel window, the filter selection window for use in determining an edge parameter, and the filter selection window for use in determining an activity metric can be of any shape and have any number of pixels to suit a particular application. Furthermore, although in this example the input pixel window and the filter selection windows are the same window of pixels, it is to be understood that the shape and number of pixels of the input pixel window and the filter selection windows can be different from each other.

One manner in which the filter selection mechanism 230 can indicate an appropriate filter that has been selected is to provide a filter identifier 236 that identifies a particular set of filter coefficients to apply to the input pixel window 210.

The filter application mechanism 240 can include a filter repository 244 for storing a plurality of filters, F\_1 to F\_N, (e.g., a plurality of sets of filter coefficients). The filter identifier 236 that is provided by the filter selection mechanism 230 can be employed to specify an appropriate filter in the filter repository 244 to apply to the input pixel window 210 to generate an output pixel.

#### Appearance Enhancement and Compressibility Improvement Mechanism 140

FIG. 3 is a block diagram that illustrates an exemplary implementation of the appearance enhancement and compressibility improvement mechanism 140 of FIG. 2. In this embodiment, a combination of an activity metric and edge parameters is utilized to select an appropriate filter for the input pixel window 210.

In this embodiment, the appearance enhancement and compressibility improvement mechanism 140 includes a level of variation measure generator 310 for receiving an input pixel, a first window of pixels (e.g., a filter selection window for use in determining an activity metric) related to the current input pixel, and based thereon for generating a level of variation (LOV) measure. The appearance enhancement and compressibility improvement mechanism 140 also includes a compare unit 320 that is coupled to the LOV measure generator 310. The compare unit 320 receives the LOV measure and a predetermined LOV measure, and based thereon determines whether the LOV measure is in predetermined relationship with the predetermined LOV measure. When the LOV measure is in predetermined relationship with the predetermined LOV measure, a blur filter or smoothing filter is provided. When applied to the input pixel window, a blur filter or smoothing filter generates a smooth value or blurred value of the current input pixel. For example, the smooth value can be an average of the pixels adjacent to the current input pixel.

When the LOV measure is not in a predetermined relationship with the predetermined LOV measure, the input pixel is processed by an edge dependent filtering.

For example, a non-linear filter may be applied to the input pixel window. For example, a sharpening filter may be applied to improve the appearance of the image in the region that has the current input pixel. The edge dependent filter selection module 340 receives a second window of pixels (e.g., a filter selection window for use in determining an edge parameter) related to the current input pixel and based thereon selects one set of filter coefficients from the plurality of sets of filter coefficients available (e.g., FILTER\_1 .. FILTER\_N). It is noted that one of the filters (e.g., FILTER\_1 .. FILTER\_N) can be a smoothing or blurring filter or an enhancement filter.

An enhancement filter can be a smoothing filter, a sharpening filter, a filter that increases contrast across an edge and smoothes along an edge, a specially designed filter, or a combination thereof. The specially designed filter, for example, can be designed to increase the contrast across an edge of a particular angle. An example of a filter that smoothes along an edge is an anisotropic diffusion filter that smoothes along an edge (i.e., blurs in an interior region). The filters can be selected from those that are well known by those of ordinary skill in the art. The filter application unit 350 applies the selected set of filter coefficients to the input pixel window to generate an output pixel.

#### Edge Dependent Filter Selection Module 340

FIG. 4 is a block diagram that illustrates in greater detail the edge dependent filter selection module 340 of FIG. 3. The edge dependent filter selection module 340 includes an edge angle measure generator 410, a sharpness measure generator 420, a curvature metric generator 430, and a filter selector 440. The edge angle measure generator 410 receives the second window of pixels related to the current pixel and generates an edge angle measure that is provided to the filter selector 440 and the sharpness measure generator 420. The sharpness measure generator 420 is coupled to the edge angle measure generator 410 for receiving the edge angle measure, and based thereon, generates a sharpness measure (SM) that is provided to the filter selector 440 and the curvature metric generator 420. The curvature metric generator 430 includes a first input that is coupled to the sharpness measure generator 420 to receive the sharpness measure, and a second input for receiving the second

window. Based on these inputs, the curvature metric generator 430 generates a curvature metric, which is provided to the filter selector 440.

The filter selector 440 is coupled to the edge angle measure generator 410, the sharpness measure generator 420, and the curvature metric generator 430 to receive the edge angle measure, sharpness measure, and the curvature metric, respectively, and based on these inputs selects one set of filter coefficients to apply to the current input pixel.

A color image typically includes a plurality of pixels that are each represented by red, green, blue (RGB) values. For example, a pixel can be represented by a total of twenty-four bits where eight bits represent the red component of the pixel, eight bits represent the green component of the pixel, and eight bits represent the blue component of the pixel. The color image is also said to include a plurality of color planes (e.g., a red color plane, a green color plane, and a blue color plane) where each color plane has the respective color components for the pixels of the image.

#### Processing Steps

According to another embodiment, the image processing method performs the following steps for each input pixel. First, an input pixel window that typically includes the current input pixel and pixels adjacent to the current input pixel is received. Second, a filter identifier is generated based on either an edge parameter computed based on the input pixel window or an activity metric computed based on the input pixel window. Third, a filter specified by the filter identifier is applied to the input pixel window to generate an output pixel corresponding to the current input pixel.

FIG. 5 is a flowchart illustrating the steps performed by the appearance enhancement and compressibility improvement mechanism in accordance with one embodiment of the present invention. In step 500, a first window of pixels (e.g., a filter selection window) is received. In step 510, the level of variation, which is an example of an activity metric, within the first window is computed. For example, the level of variation may be determined by computing the mean average deviation (MAD) of the green color plane, the red color

plane, the blue color plane, or of a plane extracted from all three color planes (e.g., a luminance plane) as described in greater detail hereinafter with reference to FIG. 6.

In step 520, a determination is made whether the level of variation computed in step 510 is in a predetermined relationship with a predetermined level of variation. For example, when the level of variation is less than a predetermined level of variation, the determination can indicate that the pixel is in a smooth uniform region that conveys little or nothing in terms of edges, texture or image detail.

In step 530, the current pixel is replaced by a blurred version of the pixel, thereby increasing the compressibility of the current region of the image. It is noted that this step may also improve image appearance.

In step 540, when the level of variation is not in a predetermined relationship with a predetermined level of variation (e.g., equal to or greater than the predetermined level of variation), this determination can indicate that the current region conveys information regarding edges, texture or image detail. In this case, an enhancement filter is selected based on edge information and applied to the input pixel window (e.g., a neighborhood of pixels that includes the current input pixel), thereby improving the appearance of the current region of the image.

#### Processing Steps for Preferred Embodiment

FIG. 6 is a flowchart illustrating the steps performed by the appearance enhancement and compressibility improvement mechanism in accordance with one embodiment of the present invention.

The input to this invention is a first input image 144, and the output is a second image 148 having the same size as the first image. When the input 144 and output 148 images are color, each pixel is defined by red (R), green (G), and blue (B) coordinate values in the range from 0 to 255, inclusive. Although an exemplary embodiment of the present invention is described hereinafter with reference to images represented in the (R,G,B) color space, it is noted that the appearance enhancement and compressibility improvement mechanism of the present invention can be applied to enhance images that are represented in

other color spaces (e.g., CIELab, CIEluv, Yuv). Furthermore, it is noted that the present invention can also be applied to improve monochrome images (e.g., grayscale images).

One output pixel is generated for every pixel in the input image. In order to determine any output pixel, it is only necessary to know a first window of pixels (e.g., a window of 5x5 pixels) centered at the corresponding input pixel. It is noted that one of ordinary skill in the art would readily appreciate that this algorithm can be modified so that it is only necessary to know a neighborhood of pixels, where the neighborhood of pixels can have any size or shape with reference to the corresponding input pixel. Output pixels may be generated in any order or even in parallel if resources allow. In this embodiment, output pixels are generated in raster order.

Referring to FIG. 6, the step 600, a mean absolute deviation is computed for each of the R, G, and B planes. The mean absolute deviations for the R, G, and B planes are denoted as rMAD, gMAD, and bMAD, respectively.

Writing the red coordinate values in the input pixel window as follows,

$$\begin{array}{ccccc}
 RI(-2, -2) & RI(-2, -1) & RI(-2, 0) & RI(-2, 1) & RI(-2, 2) \\
 RI(-1, -2) & RI(-1, -1) & RI(-1, 0) & RI(-1, 1) & RI(-1, 2) \\
 RI(0, -2) & RI(0, -1) & RI(0, 0) & RI(0, 1) & RI(0, 2) \\
 RI(1, -2) & RI(1, -1) & RI(1, 0) & RI(1, 1) & RI(1, 2) \\
 RI(2, -2) & RI(2, -1) & RI(2, 0) & RI(2, 1) & RI(2, 2)
 \end{array} ,$$

where  $R(0,0)$  is the red coordinate value of the input pixel, and the red mean absolute deviation is computed as

$$rMAD = \sum_{m=-1}^1 \sum_{n=-1}^1 |RI(m, n) - rAVE| ,$$

where rAVE is a 3x3 pixel average computed as

$$rAVE = \left\lfloor \frac{1}{9} \left( 4 + \sum_{m=-1}^1 \sum_{n=-1}^1 RI(m, n) \right) \right\rfloor ,$$



and  $\lfloor \cdot \rfloor$  denotes truncation to integer. It is noted that rMAD is actually nine times greater than a true “mean absolute deviation.” The quantities gMAD and bMAD are computed in a similar manner by using the green and blue color components, respectively.

In step 610, the AECIM 140 scales rMAD by  $\frac{1}{2}$ , and bMAD by  $\frac{1}{4}$ . This scaling step prepares rMAD, gMAD, and bMAD for comparison with each other in order to determine which color component has the greatest impact on perceived variation in the vicinity of the input pixel. Since luminance variation is a reasonable predictor of perceived color variation, the magnitudes of rMAD, gMAD, and bMAD are adjusted according to their approximate relative contributions to the luminance component.

To see that scaling rMAD by  $\frac{1}{2}$  and bMAD by  $\frac{1}{4}$  achieves the desired objective, consider that the luminance Y for an (R, G, B) pixel is often computed as

$$Y = 0.299 * R + 0.587 * G + 0.114 * B ,$$

and observe that 0.299 is approximately half of 0.587, and that 0.114 is approximately one quarter of 0.587. One desirable consequence of this scaling step is that it renders rMAD, gMAD, and bMAD all comparable to the same threshold value (e.g., THRESH1).

Next, in step 620 rMAD is compared with THRESH1. Preferably, the predetermined threshold (THRESH1) is determined empirically and remains fixed for all input images. Alternatively, the value of THRESH1 may change from image to image, or even from pixel to pixel.

If rMAD is less than THRESH1, then the red component of the output pixel (RO) is assigned to the value of the average (rAVE) of the red component (i.e.,  $RO = rAVE$ ). rAVE is computed by using a predetermined neighborhood of pixels (e.g., a second window of 3 x 3 pixels within the first window) with reference to the input pixel. In the case where rMAD is less than THRESH1, there is little or no pixel activity in the red component, thereby implying that the red component may be blurred or smoothed so as to improve compressibility without adversely affecting image appearance.

Processing steps 630 and 640 involve the same operations as described previously with reference to step 620, except that the processing is for the green plane and the blue plane.

In decision block 650, a determination is made whether any color components of the output pixel have not yet been determined. When there are color components of the output pixel that have not yet been determined, these color components are computed by using a nonlinear filter (step 660). Step 660 is described in greater detail hereinafter with reference to FIG. 7. Otherwise, the color output pixel is completely determined at this point.

#### Computing Pixel Components Using a Non-linear Filter

FIG. 7 is a flowchart illustrating in greater detail the step of computing color components by using a nonlinear filter of FIG. 6 in accordance with one embodiment of the present invention. In step 700, a filter is selected from a group of different filters based on pixels from the color component with the greatest MAD. In decision block 710, a determination is made whether a red output pixel is already determined. When the red output pixel has not yet been determined, in step 720, a red output pixel is generated by applying the filter selected in step 700 to the red plane. Otherwise, processing proceeds to decision block 730.

In decision block 730, a determination is made whether a green output pixel is already determined. When the green output pixel has not yet been determined, in step 740, a green output pixel is generated by applying the filter selected in step 700 to the green plane. Otherwise, processing proceeds to decision block 750.

In decision block 750, a determination is made whether a blue output pixel is already determined. When the blue output pixel has not yet been determined, in step 760, a blue output pixel is generated by applying the filter selected in step 700 to the blue plane.

In the nonlinear filtering process, first identify the largest among rMAD, gMAD, and bMAD (after scaling rMAD by  $\frac{1}{2}$  and bMAD by  $\frac{1}{4}$  as described above with reference to step 610). Based on the pixels in the corresponding color component, a set of filter coefficients, denoted as

$$\begin{array}{ccccc}
FC(-2,-2) & FC(-2,-1) & FC(-2,0) & FC(-2,1) & FC(-2,2) \\
FC(-1,-2) & FC(-1,-1) & FC(-1,0) & FC(-1,1) & FC(-1,2) \\
FC(0,-2) & FC(0,-1) & FC(0,0) & FC(0,1) & FC(0,2) \\
FC(1,-2) & FC(1,-1) & FC(1,0) & FC(1,1) & FC(1,2) \\
FC(2,-2) & FC(2,-1) & FC(2,0) & FC(2,1) & FC(2,2)
\end{array}$$

are selected. The filter selection process is described in greater detail hereinafter with  
5 reference to FIG. 8.

Next if the red component of the output pixel has not yet been determined, it is  
computed by applying the selected filter coefficients to the red component of the input pixel  
window. That is, the red output pixel is computed as follows:

$$RO = \sum_{m=-2}^2 \sum_{n=-2}^2 RI(m,n) \cdot FC(m,n) .$$

Similarly, if necessary the green component of the output pixel is computed as  
follows:

$$GO = \sum_{m=-2}^2 \sum_{n=-2}^2 GI(m,n) \cdot FC(m,n) ,$$

where  $GI(\cdot, \cdot)$  denotes the green input pixel values. In the same manner, if is  
necessary, the blue component of the output pixel is computed as follows:

$$BO = \sum_{m=-2}^2 \sum_{n=-2}^2 BI(m,n) \cdot FC(m,n) ,$$

where  $BI(\cdot, \cdot)$  denotes the blue input pixel values.

FIG. 8 is a flowchart illustrating in greater detail the step of selecting a nonlinear  
filter (step 700 of FIG. 7) in accordance with one embodiment of the present invention. This  
portion of the processing involves selecting a class index that corresponds to a filter designed  
for that class. There are 25 different classes, which may be divided into 4 groups. Classes 1  
20 through 8 represent relatively smooth transitions having different angular orientations. In  
this embodiment, the angles are quantized into 8 separate bins, each subtending 45 degrees

of arc. Classes 9 through 24 represent sharp edges, where classes 9 through 16 are for pixels that lie on the dark side of sharp edges, and classes 17 through 24 are for pixels that lie on the light side of sharp edges. Finally, class 0 usually corresponds to the case where there are no transitions or edges at all.

5           The class index selection is based on gradient information, which conveys the angle of any edge present at the input pixel. Depending on the gradient information, the class index selection may also be based on curvature information, which conveys the side of the edge the input pixel lies on (i.e., the darker side or the lighter side). The gradient information is represented with two coefficients,  $G_x$  and  $G_y$ ; and the curvature information is obtained based on five coefficients  $A$ ,  $B$ ,  $C$ ,  $G_x$ , and  $G_y$ . An exemplary manner in which the coefficients  $A$ ,  $B$ , and  $C$  may be computed is described in greater detail hereinafter with reference to FIG. 11.

10           In the preferred implementation, the gradient and curvature coefficients are extracted from the 5x5 window centered at the input pixel. However, it is noted that windows of different sizes (e.g., 3x3, 7x7, 9x9) and shapes can be utilized with correspondingly different operators.

15           In step 800, the gradient coefficients  $G_x$  and  $G_y$  are computed by using two operators. FIG. 9 illustrates two exemplary sets of operator coefficients 910, 920 that may be utilized in accordance with one embodiment of the present invention. The operators 910 and 20 920, respectively, find edges of orthogonal directions (e.g., the horizontal direction and vertical direction). It is noted that other sets of operator coefficients may be selected to suit a particular application.

25           In particular, the set of operators should provide comprehensive edge angle information. For example, in the embodiment described herein, it is important that the coefficients for computing  $G_x$  form an operator for finding horizontal edges. Similarly, it is important that the coefficients for computing  $G_y$  form an operator for finding horizontal edges.

For example, the operators can be a first set of operator coefficients 910 that represent horizontal "bars" where each horizontal bar features the same value and second set of operator coefficients 920 that represent vertical "bars" where each vertical bar features the same value. It is noted that the first set of coefficients 910 is tuned to find horizontal edges because the coefficients in the upper half are negative, and the coefficients in the lower half are positive. Similarly, it is noted that the second set of coefficients 920 is tuned to find vertical edges because the coefficients in the left half are negative, and the coefficients in the right half are positive.

In this example, edge angle coefficient  $G_x$  is computed by employing the following expression:

$$G_x = \sum_{m=-2}^2 \sum_{n=-2}^2 I(m,n) \cdot G_{xC}(m,n) ,$$

where  $I(\cdot, \cdot)$  represents the pixels from the color component having maximum mean absolute deviation, and  $G_{xC}(\cdot, \cdot)$  represents the first set of operator coefficients 910. Edge angle coefficient  $G_y$  is computed in a similar manner by using the second set of operator coefficients 920 at right.

Next, in step 810 it is determined whether at least one of  $G_x$  and  $G_y$  is not equal to zero. When  $G_x$  is equal to zero and  $G_y$  is equal to zero, then in step 820 the class index is defined to be zero, and the filter selection process is finished. Otherwise, when either  $G_x$  is not equal to 0 or  $G_y$  is not equal to zero, in step 830 a tentative class index between 1 and 8 is determined based on  $G_x$  and  $G_y$ . An exemplary process of determining the tentative class index is described in greater detail hereinafter with reference to FIG. 10.

In step 840, an edge sharpness metric is computed. For example, the edge sharpness metric can be computed by using the following expression:

$$G_x \cdot G_x + G_y \cdot G_y$$

In decision block 850, it is determined if the sharpness metric is greater than a predetermined threshold. When the edge sharpness metric is less than or equal to threshold

(THRESH2), then the tentative class index is regarded as the true class index, and the filter selection process is finished.

In one implementation, the value of THRESH2 is determined empirically and is intended to remain fixed for all input images. Another option would be to allow the value of THRESH2 to change from image to image or even from pixel to pixel.

When the edge sharpness metric is greater than or equal to threshold (THRESH2), then in step 860, the curvature coefficients A, B, and C are computed. A preferred process for computing the curvature coefficients A, B, and C is described in greater detail hereinafter with reference to FIG. 11. This process is similar to the procedure employed to compute the gradient coefficients Gx and Gy as shown in FIG. 9.

In step 870, a curvature metric is computed based on A, B, C, Gx and Gy. For example, the curvature metric can be computed by using the following expression that involves A, B, C, Gx, and Gy:

$$\frac{1}{\text{edge\_shrp\_metric}} \left( \frac{1}{2} \cdot A \cdot Gx \cdot Gx + B \cdot Gx \cdot Gy + \frac{1}{2} \cdot C \cdot Gy \cdot Gy \right) ,$$

where edge\_shrp\_metric is the edge sharpness metric computed above.

In decision block 880, it is determined whether the curvature metric is greater than zero. When the curvature metric is positive, then in step 890 the final class index is computed by adding eight to the tentative class index. Otherwise, in step 894 the final class index is computed by adding sixteen to the tentative class index. In either case, the filter selection process is now complete.

In the foregoing specification, the invention has been described with reference to specific embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader scope of the invention. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.